Listing of Claims

1. (Currently Amended) A network protocol processor system, the system comprising:

an interface to receive a packet;

a cache to store context data for the packet; [[and]]

a ~~processing~~ Transmission Control Protocol (TCP) offload engine to process the packet using context data in the cache; and

a Direct Memory Access (DMA) controller connected to the TCP offload engine, the DMA controller adapted to transfer data from host memory to a transfer queue and to store data from a header and data queue into host memory in parallel with the TCP offload engine processing the packet using the context data.

2. (Original) The system of claim 1, further comprising:

a working register to store the context data for a current connection that is being processed.

3. (Original) The system of claim 1, wherein the cache is capable of storing and retrieving context data for multiple connections.

4. (Original) The system of claim 1, wherein the interface comprises at least one of a host interface and a network interface.

5. (Original) The system of claim 4, wherein the host interface interacts with a doorbell queue, a completion queue, and an exception/event queue.

6. (Original) The system of claim 5, wherein each of the doorbell queue, the completion queue, and the exception/event queue is a data structure.

7. (Original) The system of claim 6, wherein each data structure has a priority mechanism.

8. (Original) The system of claim 5, further comprising:

processing logic to store the packet incoming from the host interface into the doorbell queue;

host memory to store descriptors that are pointed to by the packet;

processing logic to access the descriptors in host memory for storage in the cache; and

a scheduler to perform a hash based table lookup against the cache to correlate the packet with context data, to load the context into the working register when the context data is found in the cache, and to schedule a host memory lookup when the context data is not found in cache.

9. (Currently Amended) The system of claim 8, further comprising:

~~a Direct Memory Access (DMA) controller; and~~

processing logic to notify the DMA controller to transfer data from host memory to the transfer queue.

10. (Cancelled)

11. (Original) The system of claim 4, wherein the network interface interacts with a header and data queue and a transmit queue.

12. (Original) The system of claim 11, further comprising:

processing logic to store the packet incoming from the network interface into the header and data queue;

a working register;

a scheduler to perform a hash based table lookup against the cache to correlate the packet with context data, to load the context into the working register when the context data is found in the cache, and to schedule a host memory lookup when the context data is not found in cache.

13. (Currently Amended) The system of claim 1, further comprising:

a working register to store data for use by the ~~processing~~ TCP offload engine; and

a scheduler to locate and load the context data into the working register.

14. (Original) The system of claim 1, further comprising:

a Direct Memory Access transfer queue; and

a Direct Memory Access receive queue.

15. (Original) The system of claim 1, further comprising:

a timer; and

a hardware assist to translate a virtual address to a physical address.

16. (Currently Amended) The system of claim 1, further comprising:

a thread cache to store intermediate system state;

a core receive queue;

a working register;

scratch registers;

a ~~pieplined~~ pipelined arithmetic logic unit; and

an instruction cache.

17. (Original) The system of claim 16, further comprising:

a high bandwidth connection between the thread cache and the working register for parallel transfer of intermediate system state between the thread cache and the working register.

18. (Original) The system of claim 1, further comprising:

an instruction cache to store code relevant to specific processing, while remaining instructions are stored in at least one of host memory and cache to store context data.

19. (Original) The system of claim 1, further comprising a new instruction set including context access instructions, hashing instructions, multi-threading instructions, Direct Memory Access instructions, timer instructions, and network to host byte order instructions.

20. (Currently Amended) The system of claim 1, further comprising:

a scheduler coupled to the cache;

a working register coupled to the cache; and

processing logic in the ~~processing~~ TCP offload engine to store context data in the working register into the storage area when processing of the packet has stalled.

21. (Original) The system of claim 20, wherein the packet is a first packet and further comprising:

processing logic to load context data for a second packet from the storage area into the working register.

22. (Original) The system of claim 21, further comprising:

processing logic to restore the context data for the packet into the working register.

23. (Cancelled)

24-33. (Cancelled)

34. (Currently Amended) An article of manufacture comprising a storage medium having stored therein instructions that when executed by a computing device results in the following:

receiving a packet;

locating context data for the packet in a storage area; [[and]]

processing the packet using the context data using a Transmission Control Protocol (TCP) offload engine; and

transferring data from host memory to a transfer queue and storing data from a header and data queue into host memory with a Direct Memory Access (DMA) controller connected to the TCP offload engine, wherein the DMA controller transfers and stores data in parallel with the TCP offload engine processing the packet using the context data.

35. (Original) The article of manufacture of claim 34, wherein the instructions when executed further result in the following:

performing a lookup against the storage area to correlate the packet with context data;

loading the context into a working register in response to locating the context data in the storage area; and

scheduling a lookup of context data in response to determining that the context data is not in the storage area.

36. (Original) The article of manufacture of claim 35, wherein the instructions when executed further result in the following:

storing context data from the working register into the storage area when processing of the packet has stalled.

37. (Original) The article of manufacture of claim 36, wherein the instructions when executed further result in the following:

updating process results to the working register;

updating the storage area with the results in the working register; and

updating a thread area with the results in the working register.

38. (Original) The article of manufacture of claim 37, wherein the packet is a first packet and wherein the instructions when executed further result in the following:

loading context data for a second packet from the storage area into the working register.

39. (Original) The article of manufacture of claim 38, wherein the instructions when executed further result in the following:

restoring the context data for the first packet into the working register.